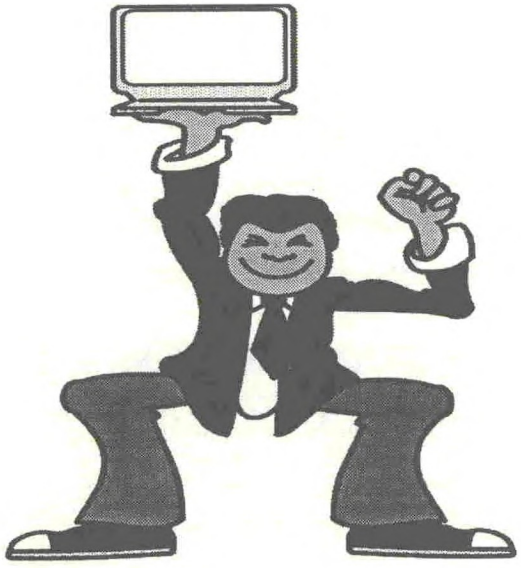
ธาวินทร์ ลีลาคุณารักษ์

SOFT MAIL

จัดทำโปรแกรมทำเนียบธุรกิจ ด้วย Visual dBASE



แนะนำโปรแกรมตัวอย่าง ที่ใช้เก็บรวบรวมรายชื่อบริษัทต่าง ๆ

ก้าจะถามว่าใครเคยได้ยินชื่อโปรแกรม dBASE หรือไม่ โดยเฉพาะคนที่เกี่ยวข้องกับคอมพิวเตอร์แทบทุกคนจะรู้จักและเคยใช้โปรแกรมชุดนี้ โปรแกรมเมอร์หลายท่านเริ่มต้นจากการสร้างโปรแกรมประยุกต์ใช้งานบน dBASE III+ เป็นโปรแกรมแรก แต่ในช่วงหลัง ๆ นี้ความนิยมในโปรแกรมชุดนี้ค่อย ๆ ลดลง สาเหตุเนื่องจากการพัฒนาโปรแกรม dBASE หยุดชะงักไปช่วงหนึ่ง ทำให้โปรแกรมลักษณะเดียวกันนี้ยี่ห้ออื่น ๆ ขึ้นมาแทนที่ แต่ถึงวันนี้ dBASE ได้พัฒนามาเป็น Visual dBASE โดยเวอร์ชันหลังสุดขณะที่ผมทำบทความชุดนี้คือเวอร์ชัน 5.5 ซึ่งจัดเป็นโปรแกรมประเภท Object Oriented Programmer แต่สำหรับตัวอย่างที่ผมนำเสนอในครั้งนี้จะเป็นตัวอย่างง่าย ๆ โดยใช้ความรู้เกี่ยวกับ dBASE เดิม ๆ ที่เราเคยใช้อยู่ โดยจะไม่เข้าไปยุ่งกับส่วนที่เป็น OOP หรือพยายามให้เกี่ยวข้องน้อยที่สุด ลองดูนะครับว่าถ้าเราเคยทำโปรแกรมประยุกต์บน dBASE III+ หรือเคยใช้ Clipper เราจะสามารถสร้างโปรแกรมประยุกต์ด้วย Visual dBASE ได้หรือไม่

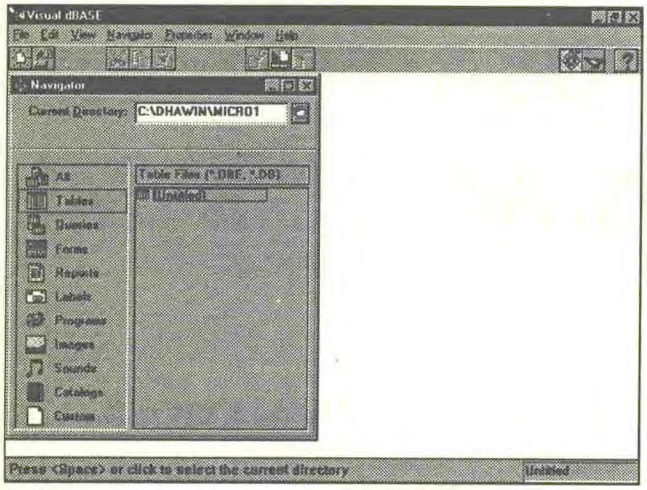
จัดสภาพแวดล้อมให้รู้จักภาษาไทย

โดยปกติโปรแกรมที่ผลิตจากบอร์แลนด์มักจะใช้ DATABASE ENGINE ของตัวเองไม่ได้ใช้ ODBC Driver ของไมโครซอฟท์ และโปรแกรมต่าง ๆ ของบอร์-แลนด์มักจะมีปัญหาเรื่องการใช้ภาษาท้องถิ่นกับตัวโปรแกรม ยิ่งถ้าเราใช้เวอร์ชันที่เป็นเบต้าแล้ว ส่วนใหญ่จะใช้ภาษาท้องถิ่นกับโปรแกรมของบอร์แลนด์ไม่ได้เลย ซึ่งจะผิดกับซอฟต์แวร์ของไมโครซอฟท์ ซึ่งจะมีปัญหาเรื่องของการใช้ภาษาถิ่นกับโปรแกรมน้อยกว่า

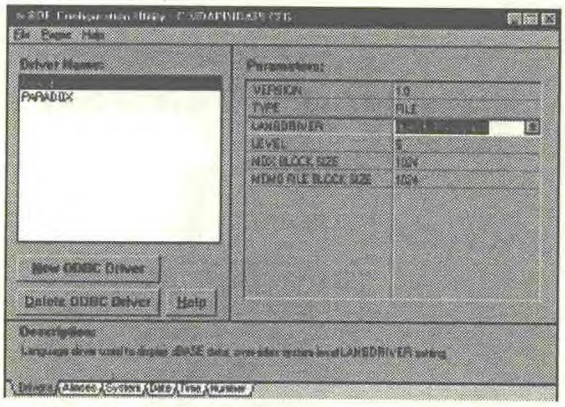
สำหรับการกำหนดสภาพแวดล้อมของ Visual dBASE ให้รู้จักภาษาไทยจะมีส่วนที่สำคัญอยู่ 2 จุด คือ

1. กำหนด LANGDRIVER ในโปรแกรม BDE Configuration Utility ซึ่งเป็นโปรแกรมควบคุมการจัดการฐานข้อมูลของบอร์แลนด์ โดยฐานข้อมูลจะมีอยู่ 2 รูปแบบ คือ ฐานข้อมูลแบบ dBASE (.dbf) และฐานข้อมูลแบบ PARADOX (.DB) และส่วนของ LANGDRIVER ก็คือการกำหนดให้ DATABASE ENGINE สามารถใช้ภาษาถิ่นในตารางข้อมูลได้ สำหรับ LANGDRIVER ที่จะใช้กับฐานข้อมูลแบบ dBASE จะใช้ DRIVER ชุด dBASE THA cp 437 ดังตัวอย่างในรูปที่ 2

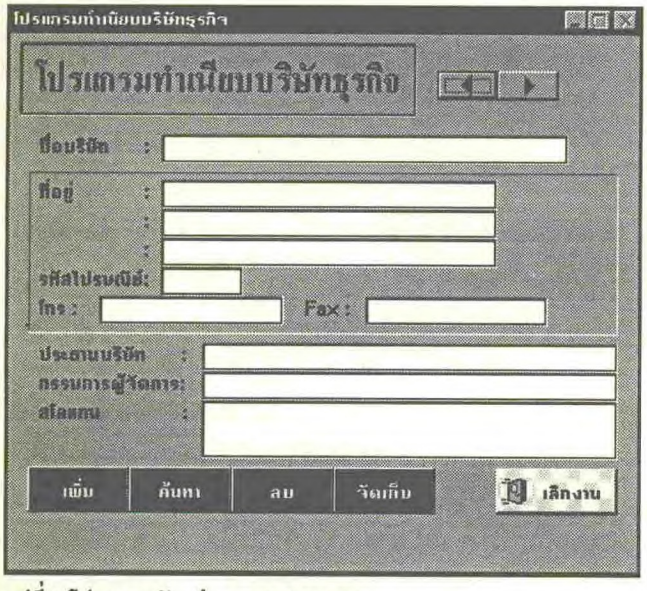
2. การกำหนดฟอนต์ของตัวอักษรที่เราจะใช้ในการเขียนชุดคำสั่งใน Editor ของ Visual dBASE โดยให้คลิ๊กที่หัวข้อ "Program" ในส่วนของ"Navigator" ของโปรแกรม จากนั้นให้คลิ๊กที่ "Untitle" ในกรอบที่อยู่ทางด้านขวาของ Navigator กดปุ่มขวาของเม้าส์แล้วเลือก "New Program" ตามลำดับ จากนั้นเลือกหัวข้อ "Properties" จากเมนูหลักของโปรแกรม แล้วเลือกหัวข้อ "Program Editor Properties" ตามลำดับ จากนั้นให้เลือกรูปแบบตัวอักษรที่สามารถแสดงภาษาไทยที่อยู่ในไดอะล็อกซ์



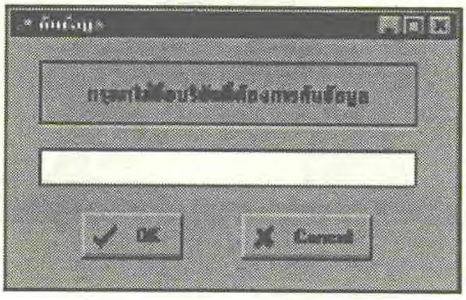
รูปที่ 1 DESKTOP ของ VISUAL DBASE



รูปที่ 2 การกำหนดภาษาอื่นในโปรแกรม BDC Configuration Utility



รูปที่ 3 โปรแกรมตัวอย่าง (TEST.WFM)



รูปที่ 4 ฟอร์มสำหรับใช้ค้นข้อมูล (XFIND.WFM)

โปรแกรมตัวอย่าง

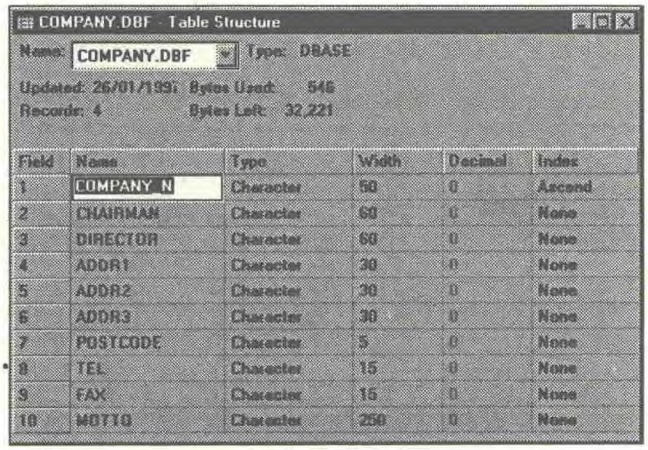
สำหรับตัวอย่างที่ผมจะใช้ประกอบบทความในครั้งนี้ เป็นโปรแกรมที่ใช้เก็บรวบรวมรายชื่อบริษัทต่าง ๆ ซึ่งผมได้ตัดมาส่วนหนึ่งจากโปรแกรมที่ใช้งานจริง เพื่อไม่ให้ตัวอย่างของเรายุ่งยากจนเกินไป โดยโปรแกรมชุดนี้จะมีฟอร์มที่เป็นหน้าจอของโปรแกรมหลัก ๆ อยู่ 2 ชุด คือ หน้าจอหลักของโปรแกรม (test.wfm) ซึ่งมีหน้าตาดังตัวอย่างในรูปที่ 3 และหน้าจอสำหรับใส่ข้อมูลที่จะใช้ในการค้นหาข้อมูลชุดที่ต้องการในรูปที่ 4 (xfind.wfm) ฟอร์มทั้งสองชุดนี้จะมีลักษณะเหมือนกับไดอะล็อกซ์คือจะไม่ขยายจนเต็มจอภาพ และฟอร์มทั้งสองชุดนี้เราจะใช้ Utility ที่ใช้สำหรับออกแบบและสร้างหน้าจอที่ Visual dBASE จัดเตรียมไว้ให้ซึ่งเราจะมาดูรายละเอียดในช่วงถัดไป

แฟ้มข้อมูลที่จะใช้กับโปรแกรม

การสร้างแฟ้มข้อมูลหรือตารางข้อมูลใน Visual dBASE จะทำได้ 2 ลักษณะ คือ ใช้พิมพ์คำสั่ง CREAT STRUCTURE ในคอมมานด์วินโดวส์ของ Visual dBASE เหมือนกับที่เราเคยใช้ใน dBASE III+ แต่พอมาเป็น Visual dBASE วิธีนี้ไม่ค่อนเป็นที่นิยมกันมากนัก

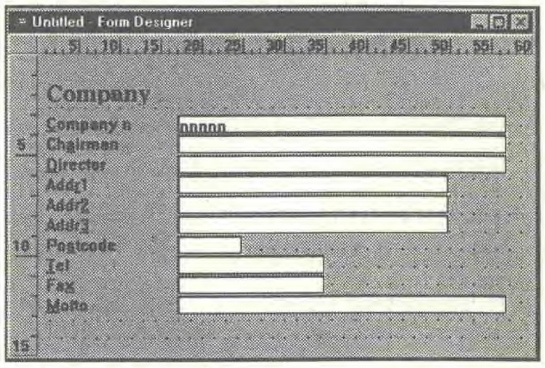
ส่วนอีกวิธีเริ่มต้นสร้างตารางข้อมูลอีกวิธีหนึ่งจะใช้ส่วน Navigator ของโปรแกรมการเขียนคำสั่ง โดยไปคลิ๊กที่หัวข้อ "Table" บน Navigator แล้วไปคลิ๊กที่หัวข้อ "Untitle" ที่อยู่ในกรอบด้านขวาของ Navigator กดปุ่มขวาของเม้าส์ แล้วเลือกหัวข้อ "New Table" ตามลำดับ สำหรับการวิธีการสร้างตารางข้อมูลของ dBASE นี้จะทำได้ 2 ลักษณะ คือใช้ Expert ช่วยในการกำหนดโครงสร้างของตารางข้อมูล หรือจะใช้วิธีกำหนดรายละเอียดของโครงสร้างตารางข้อมูลขึ้นมาเอง ซึ่งต้องขึ้นอยู่กับความถนัดของผู้ใช้แต่ละท่านนะครับ

ในรูปที่ 5 เป็นรายละเอียดของโครงสร้างแฟ้มหรือตารางข้อมูลที่เราจะใช้กับโปรแกรมประยุกต์ของเรา โดยในตารางชุดนี้ เราจะใช้ชื่อบริษัท (COMPANY_N) เป็น key หรือทำเป็น index เรียงตามตัวอักษรจากน้อยไปมาก (Ascending) ซึ่งในตอนสร้างตารางข้อมูล โปรแกรม Visual dBASE จะสร้าง Index file ขึ้นมาให้อัตโนมัติ เราไม่ต้องกังวลกับสร้าง Index file ขึ้นในภายหลัง

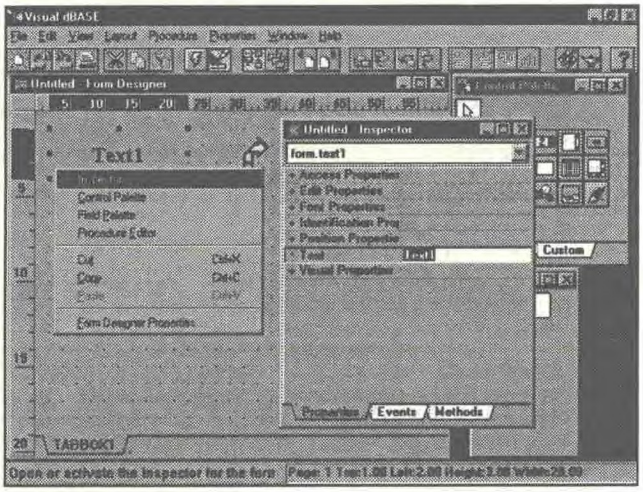


Field	Name	Type	Width	Decimal	Index
1	COMPANY_N	Character	50	0	Ascend
2	CHAIRMAN	Character	60	0	None
3	DIRECTOR	Character	60	0	None
4	ADDR1	Character	30	0	None
5	ADDR2	Character	30	0	None
6	ADDR3	Character	30	0	None
7	POSTCODE	Character	5	0	None
8	TEL	Character	15	0	None
9	FAX	Character	15	0	None
10	MOTTO	Character	250	0	None

รูปที่ 5 โครงสร้างแฟ้มข้อมูลที่จะใช้กับตัวอย่าง



รูปที่ 6 เฟรมหน้าจอที่สร้างจาก Expert ของ Visual dBASE ก่อนปรับปรุง



รูปที่ 7 ขั้นตอนการเรียกเครื่องมือต่าง ๆ

การสร้างฟอร์มหน้าจอของโปรแกรม

การออกแบบและสร้างฟอร์มหน้าจอของโปรแกรมประยุกต์บน Visual dBASE จะมี 2 ลักษณะ คือ ฟอร์มที่มีการเชื่อมโยงกับแฟ้มข้อมูลหรือตารางข้อมูล และฟอร์มที่ไม่มีการเชื่อมโยงกับตารางข้อมูล ในโปรแกรมตัวอย่างที่ผมนำมาเสนอนี้จะมีทั้งสองรูปแบบ ในรูปที่ 4 จะเป็นฟอร์มที่มีการเชื่อมโยงกับตาราง ข้อมูลที่เราสร้างเตรียมไว้ก่อนหน้านี้ ส่วนไดอะล็อกซ์ในรูปที่ 5 จะเป็นฟอร์มที่ไม่มีการเชื่อมโยงกับตารางข้อมูล

ในขณะที่เรา Run ฟอร์มหน้าจอของโปรแกรมงานที่เราจัดทำขึ้นนี้ ถ้าเป็นฟอร์มที่มีการเชื่อมโยงกับตารางข้อมูล Visual dBASE จะทำการเปิดตารางข้อมูลพร้อมกับ Index file ของตารางข้อมูลนั้น ๆ ขึ้นมาให้เราอัตโนมัติ

ขั้นตอนของการออกแบบและสร้างฟอร์ม จะเริ่มจากใช้เม้าส์คลิ๊กที่หัวข้อ "Form" ในส่วนของ Navigator ของ Visual dBASE แล้วคลิ๊กที่หัวข้อ "Untitle" ที่อยู่ในกรอบทางด้านขวาของ Navigator กดปุ่มขวาของเม้าส์แล้วเลือก "New Form" ตามลำดับ ซึ่งจะมีไดอะล็อกซ์ให้เราเลือกว่าจะใช้ Expert ช่วยในการสร้างฟอร์มหรือจะทำการออกแบบเอง (Designer) ไดอะล็อกซ์ชุดนี้จะคล้ายกับตอนเริ่มสร้างตารางข้อมูล

ในกรณีที่ฟอร์มที่เราจะสร้างนั้นมีการเชื่อมโยงกับตารางข้อมูล ผมขอแนะนำให้ใช้ Expert ช่วยจะสะดวกกว่า อย่างในตัวอย่างของเราฟอร์มที่เชื่อมโยงกับตารางข้อมูล (TEST.WFM) ผมจะใช้ Expert ช่วย generate ให้ โดยในขั้นสุดท้ายที่ Expert ทำงานเสร็จ เราจะได้ฟอร์มหน้าจอออกมาดังตัวอย่างในรูปที่ 6 ซึ่งเราสามารถปรับปรุงแก้ไขฟอร์มชุดนี้ได้ง่ายกว่าที่เราจะออกแบบเองทั้งหมด ส่วนฟอร์มอีกชุดหนึ่ง (XFIND.WFM) เราสามารถเลือก "Designer" ได้ทันที โดย Visual dBASE จะเตรียมเฟรมว่าง ๆ สำหรับให้เรากำหนดรายละเอียดลงไปเอง

และก่อนอื่นต้องทำความเข้าใจก่อนนะครับ ว่าสิ่งต่าง ๆ ที่ปรากฏอยู่บนฟอร์มหน้าจอของโปรแกรมที่เราจะทำใช้งานนี้ ตัว Visual dBASE จะมองสิ่งเหล่านี้เป็น Object ทั้งหมด ซึ่งในแต่ละ Object จะมีคุณสมบัติ (Properties) เฉพาะของแต่ละ Object และในแต่ละ Object จะมีกลไกการทำงาน (Method) ของตนเอง ซึ่งกลไกการทำงานของ Object ต่าง ๆ จะมี 2 ลักษณะ คือ กลไกหลัก (Method) เป็นกลไกที่จะทำให้แต่ละ Object มีวิธีการทำงานเฉพาะของ Object นั้น ๆ ปกติเราจะไม่สามารถเข้าไปแก้ไขกลไกในส่วนนี้ได้

แต่ในแต่ละ Object จะมีกลไกการทำงานอีกลักษณะหนึ่งคือ กลไกการทำงานตามสถานการณ์ (Event) ซึ่ง Visual dBASE จะยอมให้เราเขียนชุดคำสั่งเพื่อควบคุมการทำงานของ Object ต่าง ๆ ที่อยู่บนฟอร์ม ผ่านทางกลไกส่วนที่เป็น Event นี้ (dBASE จะจัด Event ไว้ในรูปของ Procedure ต่าง ๆ ตามเหตุการณ์ที่ควรจะเกิดขึ้นกับ Object นั้น ๆ ซึ่ง Event นี้จะแตกต่างกันไปตามชนิดของ Object)

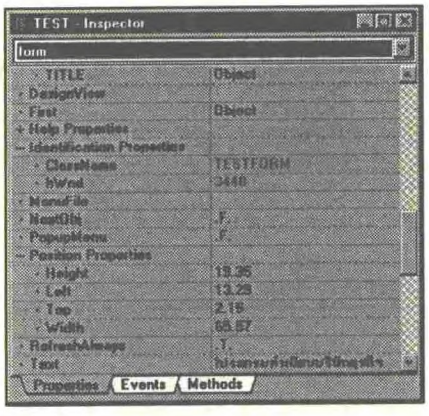
สำหรับเครื่องมือต่าง ๆ ที่เกี่ยวข้องกับการออกแบบและสร้างฟอร์มจะมีอยู่ด้วยกัน 3 ชุด คือ

1. Inspector จะเป็นไดอะล็อกซ์ที่ใช้กำหนดคุณสมบัติหลัก ๆ ของ Object ไดอะล็อกซ์ชุดนี้จะแบ่งเป็น 3 ส่วน คือ

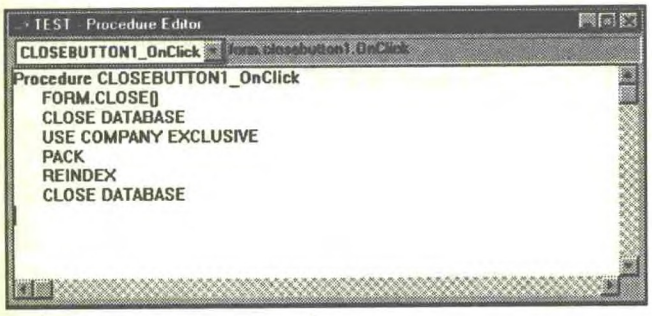
- Properties เป็นที่รวบรวมรายการของคุณสมบัติประจำตัวของ Object แต่ละแบบ ซึ่งเราสามารถปรับปรุงแก้ไขคุณสมบัติต่าง ๆ ของ Object นั้น ๆ ได้ที่ส่วนนี้ สำหรับการแก้ไขคุณสมบัติของ Object ใน Visual dBASE จะมีลักษณะบางอย่างต่างจากโปรแกรมอื่น ๆ คือเราไม่สามารถแก้ไขข้อความต่าง ๆ ที่อยู่บนเฟรมสำหรับออกแบบและสร้างฟอร์มได้โดยตรง แต่เราจะต้องทำบน Inspector นี้ ข้อความที่เราเปลี่ยนไปจะไปปรากฏบนเฟรมสำหรับออกแบบและสร้างฟอร์มอีกที่หนึ่ง

- Control Palette เป็นไดอะล็อกซ์ที่บรรจุปุ่มคำสั่งสำหรับสร้าง Object แบบต่าง ๆ เช่น เฟรม แสดงข้อความ ปุ่มคำสั่ง เฟรมแสดงรูปภาพ เป็นต้น ในไดอะล็อกซ์ชุดนี้ยังแบ่งออกเป็น 2 ส่วน คือ ส่วน Standard ซึ่งเป็นที่รวมของ Object มาตรฐานที่มีใช้ในระบบของวินโดวส์ และส่วน Custom ส่วนใหญ่จะเป็น Object เกี่ยวกับปุ่มคำสั่งที่ใช้บ่อยๆ เช่น ปุ่มคำสั่ง EXIT ปุ่มคำสั่ง OK ปุ่มคำสั่ง Cancel เป็นต้น ลักษณะที่พิเศษของปุ่มคำสั่งที่ได้จากกลุ่มของ Custom นี้คือเราสามารถนำมาใช้ได้ทันทีโดยไม่ต้องแทรกชุดคำสั่งเพิ่มเติม เพราะทางผู้ผลิตได้จัดทำ code ไว้ให้เราเรียกใช้ได้ทันที

- Procedure Editor เป็น Editor สำหรับให้เราเขียนโปรแกรมแทรกเพื่อควบคุมกลไกการทำงานของ Object



รูปที่ 8
ไดอะล็อกซ์ Inspector



รูปที่ 9 Editor สำหรับสร้างชุดคำสั่ง

ในรูปที่ 7 เป็นขั้นตอนในการเรียกเครื่องมือต่าง ๆ ขึ้นมาบนจอ โดยเราใช้เม้าส์คลิ๊กที่ Object ที่เราสนใจ แล้วกดปุ่มขวาของเม้าส์ จากนั้นให้เลือกเครื่องมือที่ต้องการจากเมนูที่ปรากฏขึ้น สำหรับรูปที่ 8 และ 9 เป็นหน้าตาของ Inspector และ Procedure Editor ตามลำดับ

หน้าจอของโปรแกรมตัวอย่างในรูปที่ 4 (TEST.WFM) นอกจากจะมีกรอบข้อความที่เป็นชื่อโปรแกรม และฟิลด์ข้อมูลต่าง ๆ ที่เกิดจากการสร้างของ Expert ของ Visual dBASE แล้ว เรายังมีการสร้างปุ่มคำสั่งลงบนฟอร์มอีก 7 ปุ่มคำสั่งด้วยกัน คือ ปุ่มคำสั่ง "เพิ่ม" ปุ่มคำสั่ง "ค้นหา" ปุ่มคำสั่ง "ลบ" และปุ่มคำสั่ง "จัดเก็บ" ปุ่มคำสั่งเหล่านี้เราใช้ Object จากเครื่องมือกลุ่ม Standard ของ Control Palette ในการจัดทำ โดยเปลี่ยนข้อความบนของปุ่มคำสั่งให้เป็นข้อความตามที่เราต้องการ

ส่วนปุ่มคำสั่ง Previous (หัวลูกศรชี้ทางซ้าย) ปุ่มคำสั่ง Next (หัวลูกศรชี้ทางขวา) และปุ่มคำสั่ง "เลิกงาน" (ต้นแบบเป็นปุ่มคำสั่ง Close) เราจะใช้ Object จากส่วน Custom ของ Control Palette

สำหรับฟอร์มในรูปที่ 4 จะประกอบด้วยเฟรมข้อความและฟิลด์รับค่า (ในที่นี้ตั้งชื่อว่า XNAME) ได้ จากส่วน Standard ของ Control Palette ส่วนปุ่มคำสั่ง OK และปุ่มคำสั่ง Cancel จะได้จากส่วน Custom ของ Control Palette ฟอร์มชุดนี้เราจะเป็นคนสร้างเองทั้งหมด

- การแทรกชุดคำสั่งและการสร้างส่วนโปรแกรมหลัก

งานขั้นถัดไปคือการแทรกชุดคำสั่งลงในจุดต่าง ๆ เพื่อควบคุมกลไกการทำงานของแต่ละ Object ให้ทำงานตามที่เราต้องการ โดยในการแทรกชุดคำสั่งลงไปใน Object ต่าง ๆ เราจะต้องเรียกส่วน Inspector ของ Object นั้น ๆ ขึ้นมาก่อน แล้วใช้เม้าส์คลิ๊กไปที่ลิ้นชีตหัวข้อ "Events" ของไดอะล็อกซ์รอไว้ก่อน เพราะในการแทรกชุดคำสั่งต่าง ๆ เราจะเริ่มที่จุดนี้

บนไดอะล็อกซ์ Inspector นี้ถ้าเราสังเกตด้านบนของ Inspector จะมีช่องบอกว่าเรากำลังทำงานบน Object ใดอยู่ ในการเปลี่ยนไปทำงานยัง Object อื่น ๆ เราสามารถใช้เม้าส์คลิ๊กที่จุดนี้แล้วเลือกชื่อ Object ที่เราต้องการจากรายการที่ปรากฏขึ้นได้โดยเราไม่ต้องเสียเวลาไปคลิ๊กที่ตัว Object ที่อยู่บนฟอร์มทุกครั้งที่เปลี่ยน Object

ในช่วงที่จะแทรกชุดคำสั่งนั้น เราจะมาทำการเลือกหาชื่อของเหตุการณ์ (Event) ที่ตรงกับเงื่อนไข การทำงานที่เราจะให้โปรแกรมทำงานบน Inspector นี้ แล้วดับเบิ้ลคลิ๊กที่ชื่อของเหตุการณ์ดังกล่าว Visual dBASE จะนำ Editor สำหรับสร้างชุดคำสั่งขึ้นมาแสดงบนจอภาพ

ในฟอร์มชุดแรก (TEST.FWM) จะมีจุดที่เราต้องแทรกชุดคำสั่งทั้งหมด 6 จุดด้วยกันโดยเรียงตาม ลำดับดังนี้

จุดแรกคือที่ตัวฟอร์ม (Form) เงื่อนไขมีอยู่ว่า ทันทีที่เราเปิดฟอร์มชุดนี้ขึ้นมาทำงาน เราจะไม่ให้ฟอร์มแสดงรายการข้อมูลที่มีการมาร์กการลบรายการไว้ จะแสดงเฉพาะรายการข้อมูลที่ใช้งานเท่านั้น ซึ่งชุดคำสั่งที่เราใช้คือ

```
Procedure Form_OnOpen &&เลือกเหตุการณ์ OnOpen
SET DELETE ON
```

จุดที่สองคือที่ปุ่มคำสั่ง "เพิ่ม" ซึ่งปุ่มคำสั่งนี้จะทำหน้าที่เพิ่มรายการว่างลงในตารางข้อมูล ชุดคำสั่งที่เราใช้มีดังนี้

```
Procedure ADDBUTTON_OnClick && เลือกเหตุการณ์ OnClick
APPEND BLANK
```

จุดที่สามคือที่ปุ่มคำสั่ง "ค้นหา" เมื่อผู้ใช้คลิ๊กที่ปุ่มคำสั่งชุดนี้ โปรแกรมจะเรียกฟอร์มชุดที่สอง (XFIND.WFM) ขึ้นมาทำงาน

```
Procedure FINDBUTTON_OnClick
DO xfind.wfm
```

จุดที่สี่คือที่ปุ่มคำสั่ง "ลบ" จะทำหน้าที่มาร์กการลบรายการข้อมูลที่ปรากฏอยู่บนจอภาพขณะนั้น ชุดคำสั่งที่ใช้มีดังนี้

```
Procedure DELBUTTON_OnClick
DELETE
```

จุดที่ห้าคือที่ปุ่มคำสั่ง "จัดเก็บ" ปุ่มคำสั่งนี้เราจะใช้ในกรณีที่มีการเพิ่มรายการข้อมูลชุดใหม่หรือมีการแก้ไขข้อมูลรายการที่อยู่บนจอภาพ ชุดคำสั่งของเราจะมีดังนี้

```
Procedure SAVEBUTTON_OnClick
IF .NOT. ISBLANK(COMPANY->COMPANY_N)
REPLACE AUTOMEM
ELSE
DELETE
ENDIF
```

สำหรับการทำงานของปุ่มคำสั่ง "ลบ" นี้ โปรแกรมจะทำการตรวจสอบว่ารายการที่จะเก็บมีข้อมูลหรือไม่ โดยตรวจสอบที่ส่วนที่เป็น Key ของตารางว่ามีข้อมูลหรือไม่ ถ้าไม่มีเราจะไม่เก็บ ส่วนคำสั่ง REPLACE AUTOMEM จะทำหน้าที่นำข้อมูลที่อยู่ในหน่วยความจำเก็บลงในตารางข้อมูล

จุดที่หกคือที่ปุ่มคำสั่ง "เลิกงาน" เนื่องจากงานชุดนี้ไม่ต้องการให้มีรายการที่ถูกลบค้างไว้ในตาราง ดังนั้นตอนเลิกงานเราก็ควรจะลบข้อมูลที่ไม่ใช้นี้ออกก่อน ชุดคำสั่งที่เราจะแทรกที่จุดนี้มีดังนี้

```
Procedure CLOSEBUTTON1_OnClick
FORM.CLOSE()
CLOSE DATABASE
USE COMPANY EXCLUSIVE
PACK
REINDEX
CLOSE DATABASE
```

ส่วนในฟอร์มชุดที่สอง (XFIND.WFM) จะมีการแทรกชุดคำสั่งเพียงจุดเดียว คือจะแทรกตรงเหตุการณ์ OnSelecttion ของตัวฟอร์ม ซึ่งเหตุการณ์นี้จะเกิดขึ้นเมื่อมีการแสดงฟอร์มบนจอภาพ แล้วผู้ใช้กด Enter 1 ครั้งเหตุการณ์ชุดนี้จะทำงานทันที

เป้าหมายของเราก็คือ เมื่อผู้ใช้ใส่ชื่อบริษัทที่เขาต้องการจะนำไปค้นข้อมูลแล้วกด Enter โปรแกรม จะนำค่าที่เก็บอยู่ Object สำหรับหรับค่า (XNAME) ไปค้นกับฐานข้อมูลทันที ซึ่งชุดคำสั่งที่เราใช้มีดังนี้

```
Procedure Form_OnSelection(nControlId)
xvalue = THIS.XNAME.VALUE
HOME()
LOCATE ALL FOR xvalue = COMPANY_N
IF .NOT. FOUND()
MSGBOX("ไม่มีข้อมูลที่ท่านต้องการในฐานข้อมูล".;
"หาข้อมูลไม่พบ")
ELSE
FORM.CLOSE()
ENDIF
```

ในชุดคำสั่งข้างต้นนี้ เราจะเห็นการเขียนคำสั่งแบบแปลก ๆ อยู่คำสั่งหนึ่งคือ

```
xvalue = THIS.XNAME.VALUE
```

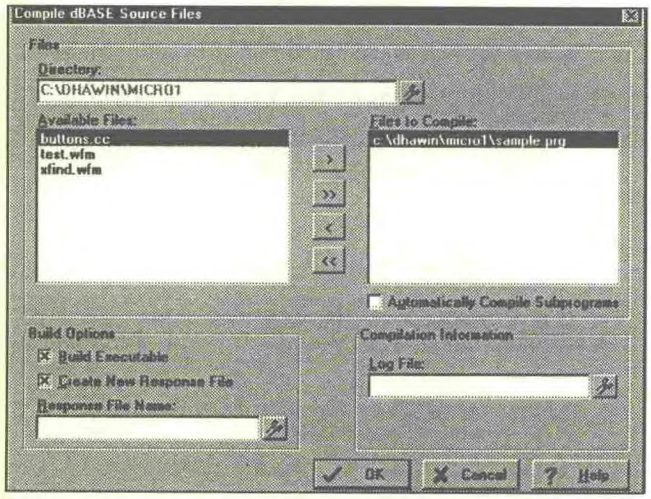
คำสั่งนี้จะทำหน้าที่ดึงค่าที่อยู่ในช่องใส่ข้อมูลที่อยู่บนฟอร์มออกมาเก็บในตัวแปรชื่อ xvalue เนื่องจากช่องใส่ข้อมูลที่ชื่อ XNAME นี้มีคุณสมบัติเป็น Object ไม่ใช้ตัวแปร ในการนำค่าที่เก็บภายในออกมาใช้จึงใช้วิธีการกำหนดค่าแบบธรรมดาไม่ได้ ต้องใช้วิธี dot notation ในการดึงค่าที่เก็บอยู่ภายในออกมา สำหรับในคำสั่งนี้ คำว่า "THIS" หมายถึงตัวฟอร์มเอง ซึ่งมีคุณสมบัติเป็น Object (ในโปรแกรมประเภท Visual ตัวอื่น ๆ จะใช้คำว่า "SELF") คำว่า "XNAME" เป็นชื่อของ Object ที่ทำหน้าที่รับค่า จากผู้ใช้ที่อยู่บนฟอร์มชุดนี้ ส่วน "VALUE" เป็นคุณสมบัติ (Property) ของ Object ชื่อ XNAME ซึ่งจะทำหน้าที่เก็บค่าต่าง ๆ ที่อยู่ภายในของ Object ชุดนี้

ข้อสังเกตอย่างหนึ่งในการสร้างชุดคำสั่งเพื่อควบคุมกลไกของ Object ต่าง ๆ ใน Visual dBASE จะมีลักษณะที่ต่างจากที่เราเคยใช้ dBASE III+ หรือ CLIPPER ตรงที่ เวลาตั้งชื่อ Procedure บน Visual dBASE เราจะไม่ได้เป็นคนตั้งเอง Visual dBASE จะดำเนินการให้เราโดยจะนำชื่อของ Object นั้น ๆ มาตั้ง แล้วตามด้วยชื่อเหตุการณ์ที่เราเลือกไว้ เช่น PUSHBUTTON_OnClick PUSHBUTTON เป็นชื่อของ Object ที่เป็นปุ่มคำสั่ง ส่วน OnClick เป็นชื่อของเหตุการณ์ที่ผู้ใช้คลิ๊กที่ปุ่มคำสั่ง

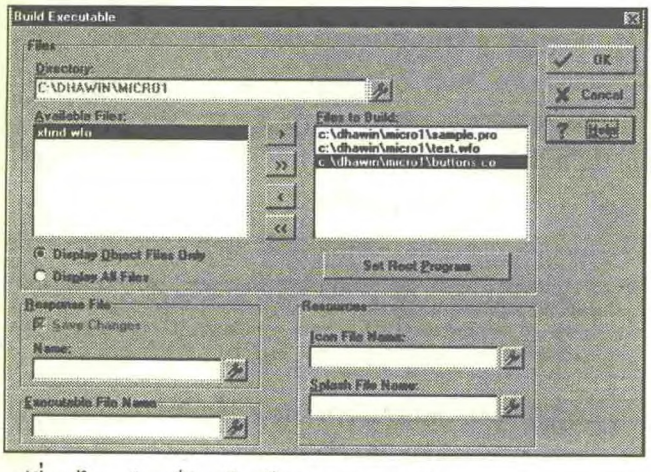
และสำหรับงานสุดท้ายที่เกี่ยวกับการสร้างชุดคำสั่งก็คือ การสร้างส่วนที่เป็นโปรแกรมหลักของงาน ซึ่งจะทำได้โดยใช้เม้าส์คลิ๊กที่หัวข้อ "Program" บน Navigator แล้วคลิ๊กที่ "Untitle" ที่อยู่ในกรอบด้านขวาของ Navigator กดปุ่มขวาของเม้าส์แล้วเลือก "New Program" ตามลำดับ Visual dBASE จะเตรียม Editor ให้เราสำหรับสร้างตัวโปรแกรม

เนื่องจากโปรแกรมตัวอย่างที่ผมนำมาเสนอนี้เป็นงานง่าย ๆ ซึ่งถ้าเป็นงานลักษณะนี้ โดยทั่วไปเราไม่จำเป็นต้องสร้างส่วนที่เป็นโปรแกรมหลักของงานเลยก็ได้ แต่เพื่อให้เป็นรูปแบบที่ดี ผมจึงได้จัดทำส่วนโปรแกรมหลักแยกต่างหาก โดยในส่วนจะมีเพียงคำสั่งเดียวคือ

```
DO TEST.WFM   && เรียกฟอร์มหลักขึ้นมาทำงาน
```



รูปที่ 10 ไดอะล็อกซ์สำหรับคอมพายล์โปรแกรม



รูปที่ 11 ไดอะล็อกซ์สำหรับสร้าง Executable file

สำหรับโปรแกรมหลักของเราจะจัดเก็บในแฟ้มชื่อ SAMPLE .PRG เพื่อเตรียมดำเนินการในขั้นตอนถัดไป

การคอมพายล์ และสร้าง Executable file

เมื่อเราเตรียมงานในส่วนต่าง ๆ ไว้เรียบร้อยแล้ว ขั้นตอนต่อไปก็คือการคอมพายล์และสร้าง Executable File เพื่อความสะดวกที่จะนำโปรแกรมไปใช้งาน สำหรับขั้นตอนการคอมพายล์จะทำได้ โดยเลือกหัวข้อ "Program" จากเมนูหลักของ Visual dBASE (หัวข้อนี้จะปรากฏเมื่อมีการเรียก editor สำหรับสร้างชุดคำสั่งขึ้นมา) แล้วเลือกหัวข้อ "Compile" ตามลำดับ ซึ่งจะปรากฏไดอะล็อกซ์ ดังตัวอย่างในรูปที่ 10 ในขั้นแรกนี้เราต้องเลือกแฟ้มต่าง ๆ ที่จะทำการคอมพายล์สำหรับงานของเราทั้งหมดก่อน โดยแฟ้มต่าง ๆ ที่เราต้องคอมพายล์มีดังนี้

- SAMPLE.PRG โปรแกรมหลักของงาน
- TEST.WFM ฟอร์มหลักของโปรแกรม
- XFIND.WFM ไดอะล็อกซ์สำหรับค้นข้อมูล
- BUTTON.CC เก็บรายละเอียดของปุ่มคำสั่งแบบต่าง ๆ ของ Visual dBASE

สำหรับ BUTTON.CC เป็นแฟ้มข้อมูลที่ทางบอร์แลนด์เตรียมไว้ให้จะอยู่ในซับไดเรคทอรีชื่อ \customer ในไดเรคทอรีของ Visual dBASE เราต้องคัดลอกแฟ้มนี้มาไว้ที่ไดเรคทอรีที่เราทำงานอยู่ก่อนที่จะทำการคอมพายล์

และหลังจากที่ Visual dBASE ทำการคอมพายล์งานเสร็จ ถ้าเรามีส่วนที่สร้าง Executable file อยู่จะปรากฏไดอะล็อกซ์ดังรูปที่ 11 ขึ้นมาอีกชุดหนึ่ง ซึ่งจะใช้สำหรับรวมงานทั้งหมดเป็น .EXE ถ้าเราต้องการใส่ ICON ลงในโปรแกรมของเราก็สามารถกำหนดลงไปในไดอะล็อกซ์นี้ก่อนที่จะทำการสร้างเป็น Executable file

ในซอฟต์เมล์ที่ผมเตรียมไว้ให้นั้น ผมไม่ได้รวมส่วนของ BORLAND DATABASE ENGINE ไปให้ด้วย

ถ้าเครื่องของท่านไม่มี BORLAND DATABASE ENGINE อยู่ โปรแกรมชุดนี้จะทำงานไม่ได้ แต่ถ้าท่านมี Visaul dBASE อยู่แล้ว ก็ไม่มีปัญหาอะไร

สำหรับความยากในการใช้โปรแกรมประเภท Visual มาสร้างโปรแกรมประยุกต์ใช้งานต่าง ๆ จะอยู่ที่การเลือก Event ของ Object ต่าง ๆ ที่เราต้องการแทรกชุดคำสั่งให้โปรแกรมทำงานตามที่เราต้องการ ถ้าหากเราเคยเขียนโปรแกรมด้วยวิธีการสร้าง code มาบ้าง การเลือก Event ที่จะนำมาใช้ก็จะทำได้ง่ายขึ้น เนื่องจากเราพอจะทราบแล้วว่างานแต่ละจุดมีแนวคิดการทำงานอย่างไร แต่ถ้าเราไม่เคยเขียนโปรแกรมมาก่อนเลย ก็ต้องลองผิดลองถูกกันพอสมควรครับ เรื่องของ dBASE ที่ผมนำมาเสนอในครั้งนี้เป็นตัวอย่างง่าย ๆ เชื่อว่าคงจะเป็นแนวทางสำหรับท่านที่สนใจโปรแกรมชุดนี้ตามสมควรนะครับ หากผมมีโอกาสจะพยายามนำเรื่องของ dBASE มาเสนอแด่ท่านผู้อ่านอีก พบกันใหม่โอกาสหน้าครับ